A Hybrid CNN and Segmentation-Based Pruned Deep Learning Approach for Precision Plant Disease Detection

Mohana Saranya Sellappan
mohanasaranya.cse@kongu.edu
Kongu Engineering College
Rajesh Kumar Dhanaraj
Symbiosis International (Deemed University)
Lalitha Krishnasamy
Nandha Engineering College
Dragan Pamucar
Szechenyi Istvan University

Article

Keywords: Deep learning, CNN, Plant disease classification, Image segmentation, hybrid deep learning models, UNet, DeepLabV3+, Model Pruning

Posted Date: May 9th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6505515/v1

Additional Declarations: No competing interests reported.

A Hybrid CNN and Segmentation-Based Pruned Deep Learning Approach for Precision Plant Disease Detection

Mohana Saranya Sellappan*[1], Rajesh Kumar Dhanaraj[3], Lalitha Krishnasamy[2], Dragan Pamucar[4]

[1]Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, Erode, Tamil Nadu, India.

[2]Department of Artificial Intelligence and Data Science, Nandha Engineering College, Perundurai, Erode, Tamil Nadu, India.

[3]Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, India

[4] Széchenyi István University, Győr, Hungary

mohanasaranya.cse@kongu.edu, sangeraje@gmail.com, vrklalitha24@gmail.com, pamucar.dragan@sze.hu

ABSTRACT

The identification of plant diseases has become increasingly challenging due to the interference of complex backgrounds in images, which often hinders the accuracy of classification models. Recent studies have employed various Deep Learning (DL) techniques to overcome this issue, utilizing both publicly available and custom datasets. However, achieving high accuracy while managing background complexity remains a significant hurdle. This paper aims to address this challenge by introducing a two-step DL approach for plant disease classification. The approach begins with an enhanced Convolutional Neural Network (CNN), developed through a comparative analysis of several CNN architectures, including customized and cascaded versions of prominent DL models, achieving an accuracy of 93.3%. To further enhance accuracy, segmentation techniques such as DeepLabV3+, UNet, Iterative UNet, and UNet with Atrous Spatial Pyramid Pooling (ASPP) are integrated before applying customized CNN architectures. These segmentation methods effectively isolate diseased portions of leaf images, improving classification performance. The proposed methodology introduces model pruning to optimize performance and computational efficiency by removing redundant parameters and less significant features. The UNet with ASPP architecture, in combination with pruning strategies, significantly reduces time complexity and feature redundancy, leading to an impressive accuracy of 99.8%. This approach outperforms other existing models in terms of accuracy and efficiency. The model is trained on the Plant Village dataset, which includes 10 different diseases across plant species such as tomato, corn, and potato, offering a comprehensive solution for plant disease identification.

Keywords: Deep learning, CNN, Plant disease classification, Image segmentation, hybrid deep learning models, UNet, DeepLabV3+, Model Pruning

1. INTRODUCTION

Plant diseases deteriorate the quality of plants and it affects the growers of broad-acre crops. The contagious agent in plants can be viral, nematodes, bacterial, or fungal. This can impair the plant parts underneath or over the ground. Finding out the traits of the infected region and knowing when and how to successfully control diseases is an occurring challenge in the agricultural field. Tomato, Corn and Potato are some of the significant food crops across the globe. It has a per capita utilization of 20 kilograms each year and addresses around 15% of the average total vegetable utilization. Potato and tomato top vegetable choices across countries like the USA. China, India, and the USA remain are top consumers of Corn. To fulfill the need for these crops it is fundamental to devise methods to work on the yield of tomato, potato and corn and advance early discovery of pests, bacterial, and viral infections. The tomato, potato and corn crops are harvested mainly during the winter and summer seasons. Because of all these environments, plants become exceptionally open to diseases caused by viruses, bacteria, and fungi. Spotting plant disease is one of the significant research topics [1].

The rapid transmission of diseases across borders has become more prevalent than ever, adding a layer of complexity to the current situation. Mitigating the harm inflicted by diseases during plant growth underscores the importance of proactive crop disease prevention measures. Recognizing the urgency of this need, there is a compelling call for the implementation of automated image-based tools. In the domain of research, DL [2] shows great promise, especially in improving accuracy compared to traditional Machine Learning (ML) models commonly used for identifying and classifying crop diseases. The notable advancements achieved by DL in computer vision have notably expanded possibilities for early detection of plant diseases, representing a revolutionary stage in agricultural methodologies.

DL represents a promising breakthrough in research, particularly in its ability to enhance accuracy compared to conventional ML models traditionally employed for identifying and classifying crop diseases. Recent advancements in DL within the field of computer vision have significantly broadened possibilities for the early diagnosis of plant diseases, marking a transformative phase in agricultural practices. While traditional ML models have their merits, they encounter challenges in managing the complexity and variability inherent in agricultural datasets, especially those related to plant diseases. In contrast, DL excels at extracting intricate patterns and features from images, enabling more nuanced and precise detection and classification of diseases affecting crops.

DL represents a promising breakthrough in research, particularly in its ability to enhance accuracy compared to conventional ML models traditionally employed for disease classification. DL within the field of computer vision have significantly broadened possibilities for the early diagnosis of plant diseases, marking a transformative phase in agricultural practices. While traditional ML models have their merits, they encounter challenges in managing the complexity and variability inherent in agricultural datasets, especially those related to plant diseases. In contrast, DL excels at extracting intricate patterns and features from images, enabling more nuanced and precise detection and classification of diseases affecting crops.

DL is a learning system based on neural networks. One notable advantage of DL is its ability to autonomously extract features from images. In recent years, CNN-based DL models have found extensive applications in computer vision, such as medical image recognition like gastric cancer [3], Diabetic Retinal Blood Vessel Images [4] and segmentation of epidermis in high-frequency ultrasound of pathological skin [5], expression recognition [6][7], traffic sign detection [8][10], complex traffic detection [9], text detection [11][12], face recognition, and more. Traditional image classification and recognition methods, as discussed in [13], are limited to extracting basic features and struggle with capturing extraordinary and complex image feature data. DL techniques, on the other hand, address and overcome this limitation.

CNN possesses a robust feature extraction capability, which has been leveraged by numerous researchers in prior studies focusing on plant disease prediction methods utilizing DL. This technique has been seamlessly integrated with conventional classifiers. Semantic segmentation method using the U-net model for Pest Detection and Identification on Tomato Plants [14], severity of foliar symptoms [15] ,corn leaf disease detection [16], DeepLabV3+ for black gram disease classification [17], grape disease identification [41], apple leaf disease [42], Iterative UNet and UNet with ASPP is performed which helps in segmenting the image. The segmented images serve as input for a pre-trained CNN network to obtain features characterizing the images. ResNet [18][19] is a new neural architecture for reducing complexity, which allows us to train extremely Deep Neural Networks (DNN). It allows us to solve the degradation problem while keeping good performance. By reducing complexity, a smaller number of parameters need to be trained and less time on training as well. AlexNet is one of the leading architectures that can be used for object detection [20] and has been applied to many Artificial Intelligence (AI) problems. In addition to pre-trained models a customized model was constructed with convolutional pooling layers and batch normalization.

The results of the experiments undeniably illustrate the efficacy of the segmentation, feature extraction, and concatenation processes employed in the models proposed. To succinctly capture the noteworthy contributions, it is delineated as follows:

- The study introduces two novel segmentation architectures—Iterative UNet and UNet with ASPP module—to enhance the accuracy of diseased region identification in plant leaf images. These architectures are systematically compared with established segmentation models, demonstrating their superior performance in delineating and isolating affected areas.
- A unique feature extraction methodology is proposed, integrating customized architectures with predefined models to enhance feature representation. The hybrid approach ensures the extraction of highly discriminative features, leading to improved classification accuracy in plant disease identification.
- To optimize computational resources, weight-based pruning techniques are incorporated into the proposed architectures. This enhances the efficiency of the models by reducing computational complexity while maintaining segmentation and classification accuracy, making the approach feasible for real-time agricultural applications.
- The study highlights the practical implications of the proposed methodologies by evaluating their performance across various plant disease datasets, ensuring robustness and adaptability in real-world agricultural scenarios.

The following sections provide an overview of the literature surveyed in this area. Section 3 details the materials and methods employed, while Section 4 presents the results and discussion. Section 5 compares the proposed work with existing state-of-the-art methods, and Section 6 discusses the conclusions and future scope.

2. LITERATURE SURVEY

2.1 Plant disease segmentation

Yonghua Xiong et al. proposed an algorithm to introduce a detection technique for cash crop diseases. In order to detach the image background without human intervention an Automatic Image Segmentation Algorithm that uses the GrabCut algorithm is used but it retains the disease spots in cash crops. It was able to give a maximum recognition rate of over 80% for the 27 diseases [22]. The Background of the infected regions is separated by the coefficient-based segmentation which was proposed by Muhammad Attique Khan et al., in [23]. Then VGG16 and caffeAlexNet pre-trained models were used to extract features from chosen disorders. To fuse the extracted features before the max-pooling stage, a parallel features fusion step is integrated. Before using multi-class SVM, the majority of univariant characteristics are selected using genetic algorithms. Openly available datasets such as- plant village and CASC-IFW are used for experiments to reach the 98.60% classification accuracy target.

In a study conducted by T. Daniya et al., they employed segmentation techniques using the Segmentation Network (SegNet). The integration of statistical, CNN, and Texture features was a key facet of their methodology. These distinctive features, when harnessed within the framework of Deep Recurrent Neural Networks (RNN), were utilized for predictive modeling of plant diseases. To optimize the Deep RNN's performance, the researchers employed the RideSpiderWater Wave (RSW) algorithm during the training phase. It gave an accuracy rate of 90.5%. This underscores not only the efficacy of the Deep RNN approach but also highlights the superior performance facilitated by the synergistic integration of SegNet segmentation, statistical, CNN, and Texture features, all guided by the innovative RSW algorithm [24].

Shuo Chen et al., proposed a new method for BLS and Rice leaf lesion detection. The Segmentation was performed using the UNet network. In order to improve the precision of lesion segmentation, BLSNet used a consideration mechanism and multi-scale extraction integration. The proposed BLSNet model demonstrated higher segmentation and class accuracy, with a score of 98.2% [25].

Md. Arifur Rahman and his research team executed a sophisticated segmentation approach, incorporating a sequential application of thresholding and morphological operations. To bolster the classification process, they employed a DNN. The noteworthy outcome of this research was a remarkable accuracy rate of 99.25%. It is crucial to highlight that their study relied on the comprehensive PlantVillage dataset [26], emphasizing the robustness and reliability of their findings within the context of plant disease diagnosis and classification.

Yahya Altuntas and colleagues introduced a method for deep feature extraction to identify Tomato Leaf Diseases. Established pre-trained models like AlexNet, GoogLeNet, and ResNet-50 were employed as feature extractors. The deep features extracted from these models were combined to enhance prediction performance. The SVM classifier underwent training using 1000 deep features obtained from the final fully connected layers of the CNN models. The accuracy rating for these combined deep features was 96.99%. The PlantVillage Dataset was used [27]. For feature extraction using PlantVillage, Faye Mohameth et al., assessed the CNN models have to do with transfer learning and deep feature extraction. Three different CNN models were utilized, for example, ResNet 50, Google Net and VGG-16 two unique classifiers SVM and KNN [28].

ResNet is a well-known pre-trained CNN architecture, Vinod Kumar et al., worked on an open Dataset which consists of 15200 crop leaves images to perform a task of feature extraction and classification using ResNet34 model. The proposed model achieved a 99.40% accuracy on a test set [29]. Malliga Subramanian et al. utilized several pre-trained models to extract features and classify corn leaf diseases. The application of these trained models resulted in an accuracy of 93% in the classification of corn leaf diseases. The PlantVillage Dataset [30] was employed for this purpose.

Verma et al., have carried out three pretrained models for detecting severity of tomato late blight disease. The tomato late blight leaf images were taken from Plant Village data set based on three categories of severity namely- early, middle and end. The AlexNet, InceptionV3 and SqeezeNet pre-trained models were used for extracting features and extracted features were used for training multiclass SVM. In the proposed work AlexNet was able to perform well with highest accuracy of 93.4% [32].

Chunshan Wang et al., suggested a two-stage model for cucumber leaf disease severity classification (DUNet) on complicated backdrops that merges DeepLabV3+ and U-Net. The DeepLabV3+ Model first segments the complicated backgrounds. The images are utilised as the input for the second step after the background has been segmented. UNet is employed for the segmentation in the second stage. It has a Dice coefficient of 0.6914 and a 93.27% accuracy rate. The average disease severity accuracy has been 92.85% [33].

In order to tackle the issue of varying size of maize disease, Yong Yang in [39] introduces an enhanced segmentation network model called Deeplabv3+. The proposed model addresses the problem through two main stages. Firstly, in the encoding stage, Resnet101 with Atrous convolution is employed for feature extraction, along with the inclusion of the Jump-Connected Atrous Spatial Pyramid Pooling (JCASPP) module. This combination allows for the acquisition of multi-scale semantic information. Secondly, in the decoding stage, the output of JCASPP is merged with the shallow features of the backbone network. This integration facilitates the enrichment of spatial information through the utilization of multilayer and small multiplicative up sampling techniques.

In [43], Surajit Mandal discussed recent advancements and challenges in the field of plant leaf disease detection. Utilizing advanced imaging methods and DL, the survey aims to provide a valuable reference for researchers involved in plant disease detection. It presents cutting-edge models to optimize the detection process, reducing time and costs. Additionally, the article explores existing challenges in the detection process, emphasizing the importance of finding solutions.

2.2 Segmentation in diverse applications

Khaoula Belhaj Soulami et al. introduced a UNet model designed to identify, segment, and predict breast cancer. The segmentation and classification achieved an Intersection over Union (IOU) score of 90.5%, and the curve reached 99.88% when applied to DDSM and INbreast datasets. The technique involved classifying every pixel in mammogram images. The utilized datasets encompassed a Digital Database for

Screening Mammography (DDSM), a Curated Breast Imaging Subset of DDSM (CBIS-DDSM), and INbreast [21].

Mobeen-ur-Rehman et al. implemented the computerized tools to recognize Diabetic Retinopathy. They have used CNN models that are pre-trained specifically AlexNet, SqueezeNet and VGG16 for classification that gives the accuracy of 93.46%, 91.82% and 94.49% discreetly. Additionally, a modified CNN model with five layers, two convolutional layers and three fully linked layers is suggested. Sensitivity, accuracy, and specificity for this approach were optimistic, with values of 98.94%, 97.87%, and 98.15%, respectively. They used the MESSIDOR Dataset [31].

Bandar Alotaibi and MunifAlotaibi implemented hybrid architecture by combining Inception and Resnet architectures for hyperspectral image classification (HSI). Four standard HSI datasets were utilised to test the model. The proposed model was expert to achieve the highest of 99.02% accuracy on the Pavia Centre scenes dataset [34]. The classification accuracy largely relies on depth of network and so Yang Li et al., made a deep feature fusion model which uses pre-trained ResNet50 and VGG16 models to extract features. In the proposed work they made use of the UC Merced land-use dataset with 21 classifications and NWPU-RESISC45 dataset with 45 scene classes. The model was able to attain 99.31% with UCMD dataset and has obtained 94.03% with NWPU45 dataset [35].

The main drawbacks discussed are the heavy model architectures and the enormous features accumulated for image processing.

- Conventional models suffer from excessive computational overhead, as they rely on complex and heavy architectures, leading to inefficiencies in image processing tasks.
- The accumulation of an overwhelming number of features in image processing reduces model efficiency, making them challenging to manage and optimize.
- There is a pressing need to explore hybrid segmentation approaches, such as integrating UNet with other segmentation algorithms, to enhance segmentation accuracy while reducing computational complexity.
- Existing studies primarily leverage features from pre-trained models like VGG16, AlexNet, and ResNet, but more advanced feature integration

techniques are required to further improve classification accuracy and efficiency.

So in this work the following approaches are proposed,

- The heavy models are replaced with a light weight customized model like Deeplab V3+, UNet and its variants for segmentation to reduce the total trainable parameters.
- Customized Efficient net model is used to improve classification accuracy.
- Pruning concept was included to reduce model complexity.

3. MATERIALS AND METHODS

3.1 Dataset Description

For plant leaf disease detection, tomato, potato, and corn plant leaves are considered primarily here and the dataset taken from www.kaggle.com. The leaf images come from the famous plant village dataset with the total size of 15079 images. Table 1 represents the list of plant leaf diseases with their respective number of images that are used in the dataset.

Table 1 Representation of the Dataset

Diseases	Images
Corn Cercospora leaf spot/ Gray leaf spot	513
Corn Common rust	1192
Corn Northern Leaf Blight	985
Corn healthy	1162
Potato Early blight	1000
Potato Late blight	1000
Potato healthy	152
Tomato Bacterial_spot	2127
Tomato Yellow_Leaf_Curl_Virus	5357
Tomato healthy	1591
TOTAL	**15079**

In Fig 1, a comprehensive illustration is presented, showcasing a representative sample set featuring both diseased and healthy images sourced from the plant leaf dataset[50]. This dataset is characterized by a substantial compilation, encompassing a total of 15,079 images capturing the diverse spectrum of healthy and unhealthy states across tomato, potato, and corn crops.

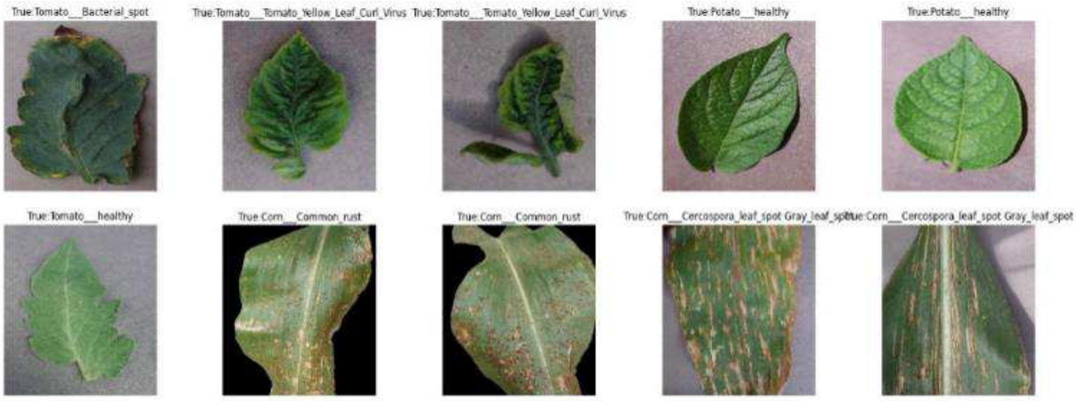


Fig.1 Plant Village dataset sample images

The dataset further stratifies these images into 10 distinct classes based on their unique characteristics as shown in figure 1. Within this extensive dataset, a judicious allocation is made for training, validation, and testing purposes. Specifically, a significant subset of 13,554 images is earmarked for training the model, ensuring its robust learning from diverse instances. Additionally, 1,505 images are designated for the validation dataset, facilitating the fine-tuning and optimization of the model's performance. Finally, a succinct yet crucial set of 20 images is reserved for the testing dataset, serving as the litmus test for the model's predictive prowess in real-world scenarios. This meticulous distribution underscores the methodical approach undertaken to enhance the model's accuracy and efficacy in handling diverse instances of plant diseases across multiple crop types.

3.2 Pre-processing

Image pre-processing plays a crucial role in transforming the raw dataset into a format suitable for network analysis. Given that images in the dataset vary in size and come from different sources, including those with noisy backgrounds, the initial step involves resizing them to a standardized dimension. To achieve this, a resizing function is employed, ensuring that 256x256-pixel square images are effectively transformed into 224x224-pixel square images. This process carefully maintains the aspect ratios of the images while utilizing zero padding, a strategy that aligns with the model's requirements and contributes to improved results.

4. Proposed Methodology Proposed Semantic Segmentation

a. System Model

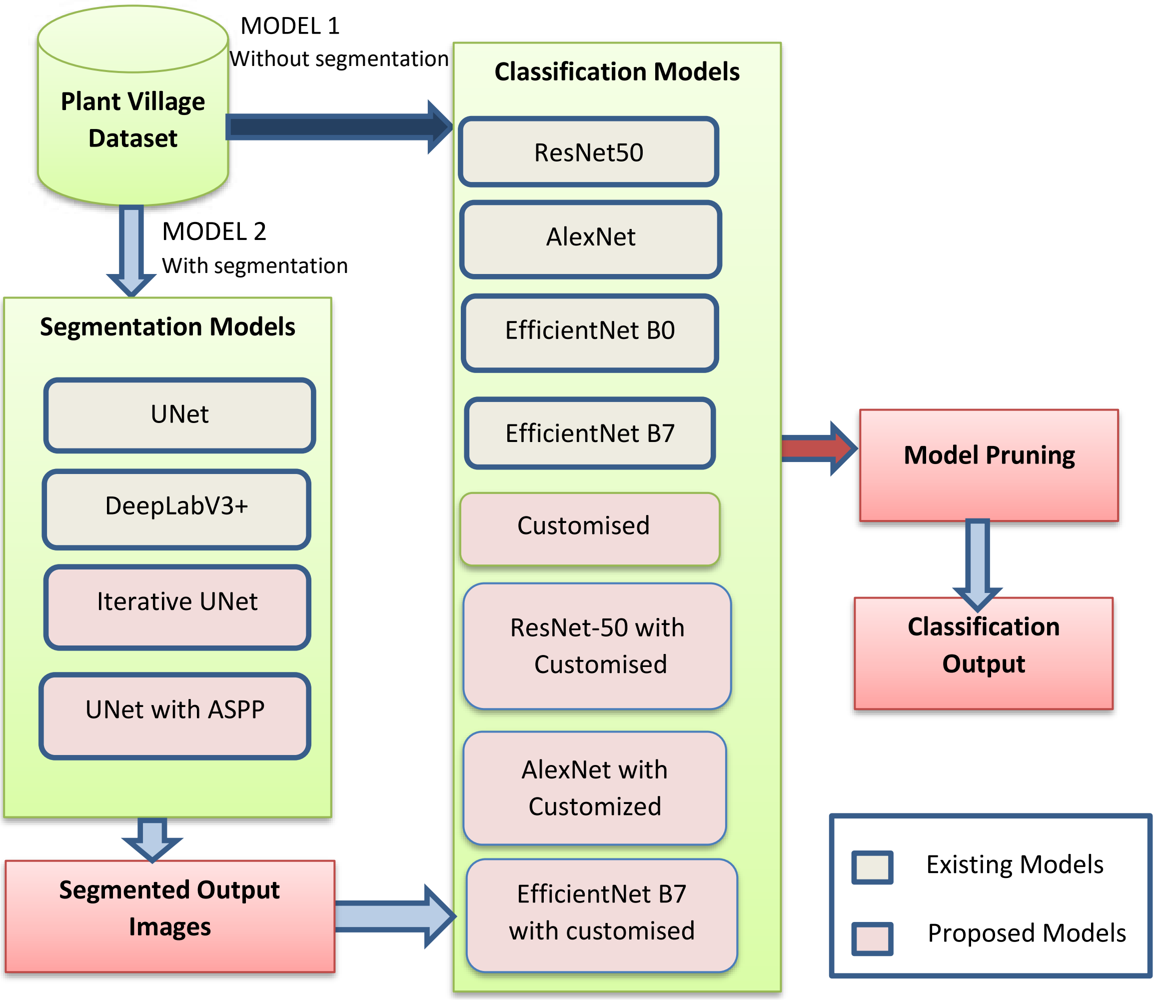


Fig.2 Proposed Workflow

Semantic Segmentation involves assigning a class label to each pixel in an image, distinguishing it from classification, where a single class is assigned to the entire image. In semantic segmentation, instances of the same class are treated as individual objects, while instance segmentation considers different instances of a similar class as distinct items. Following preprocessing, the image undergoes segmentation, dividing it into segments with similar intensities and similarities as clearly depicted in figure 2. Subsequently, the image enters the Hue Saturation and Intensity (HIS) model. Semantic segmentation is executed using the U-net architecture, an extension of CNNs with some modifications. This sophisticated model effectively addresses existing challenges, demonstrating faster image segmentation capabilities, processing images of size 512*512 within seconds. In reference to [36], images pass through an encoder where spatial information decreases and feature information increases, followed by the same passing through a decoder. Post-segmentation, the images are subjected to a CNN model for feature extraction.

b. ***Proposed Iterative UNet Architecture***

The architecture used in this image segmentation model is Iterative U-Net architecture.

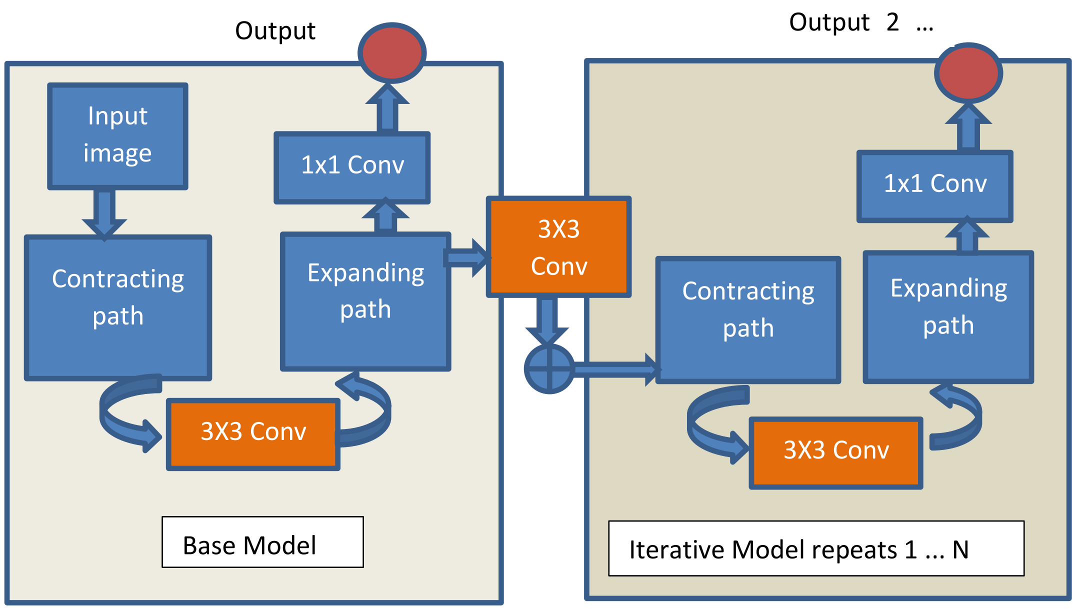


Fig.3 Iterative U-Net Architecture

The fundamental structure of U-Net consists of a contracting and expanding path, giving it a U-shaped configuration and earning it the name U-Net. This architecture is primarily constructed as an end-to-end fully convolutional network [37]. Illustrated in Figure 3 is the diagram of the Iterative UNet architecture, depicting the iterative application of the U-Net structure in both encoding and decoding. This iterative process results in a U-shaped pattern and provides a comprehensive view of the convolutional and pooling layers involved. The UNet structure is applied iteratively for N-1 iterations, where an increase in N leads to a proportional increase in trainable parameters. This introduces a tradeoff between time, space complexity, and performance.

The encoder part of the iterative UNet architecture consists of downsampling blocks, similar to the traditional UNet. Each downsampling block typically includes convolutional layers followed by pooling or strided convolutions to progressively reduce the spatial resolution while increasing the number of feature channels. This helps in capturing context and extracting high-level features. The output of each convolutional layer is given by Equation 1,

$$Z_l = \sigma\,(W_1 * A_{l-1} + b_l) \tag{1}$$

where Z_l is the output, W_l is the weight matrix, A_{l-1} is the input from the previous layer, b_l is the bias term, and σ is the activation function.

Instead of a single decoder, the iterative UNet incorporates multiple decoder blocks, each refining the segmentation results iteratively. Each decoder block takes the output of the previous decoder block and features from the corresponding encoder block. The output of each upsampling layer is given by Equation 2,

$$U_l = Upsample(Z_{l-1}) \tag{2}$$

where U_l is the upsampled output. Concatenation of the upsampled output with the corresponding feature map from the encoder will be denoted in Equation 3 as,

$$C_l = Concatenate(U_l, Z_{encoder}, l-1) \tag{3}$$

Skip connections are formed by concatenating feature maps from the encoder to the decoder to preserve spatial information. The final layer typically involves a 1x1 convolution to produce the segmentation output.

$$Y = \sigma(W_f * C_{final} + b_f) \tag{4}$$

Where Y is the final output which is shown in Equation 4. This iterative process helps in refining the segmentation progressively, allowing the model to capture more fine-grained details and improve accuracy.

c. Proposed UNet Architecture with Atrous Spacial Pyramid Pooling module

The UNet architecture has been augmented with the ASPP module from the DeepLab V3+ variant. This integration involves placing the ASPP module before the decoder, facilitating the capture of multi-scale contextual information. The ASPP module comprises parallel atrous/dilated convolutions with distinct dilation rates, strategically employed to grasp context at varying resolutions. The resultant feature maps from these parallel convolutions are subsequently merged to form a more comprehensive representation of the input. By incorporating the ASPP module, the architecture effectively captures both local and global contextual information, thereby enhancing segmentation performance.

Expressing the U-Net architecture with ASPP mathematically involves a series of equations. For clarity, a conceptual representation of the U-Net with ASPP is presented here. It is important to note that this representation offers a high-level overview, and actual implementations may entail additional details and considerations. Let X denote the input image and Y represent the output segmentation mask. The U-Net with ASPP can be dissected into encoding and decoding stages, with the ASPP module typically integrated into the encoding stage to ensure the capture of multi-scale contextual information.

Encoding stage: U-Net involves down-sampling and capturing features at different scales. Input Convolution can be denoted as in Equation 5,

$$F_1 = Conv\ (X, W_1) + b_1 \tag{5}$$

where Conv represents the convolution operation, W_1 is the weight matrix, and b_1 is the bias term. Down sampling can be denoted as in Equation 6,

$$D_1 = MaxPool(F_1) \tag{6}$$

where MaxPool is the max pooling operation.

ASPP module: can be denoted as Equation 7 and Equation 8,

$$A_i = Conv_{dilated}(D_1, W_{asppi}) \quad (7)$$

$$A = Concat(A_1, A_{2,}A_{3,\dots}A_n) \quad (8)$$

where $Conv_{dilated}$ is the dilated convolution operation, W_{asppi} represents the weights for each dilated convolution, and $Concat$ denotes the concatenation operation.

Decoding stage:

The decoding stage involves up-sampling and combining features from the encoding stage. Upsampling will be denoted as in Equation 9,

$$U_1 = UpSample(A) \quad (9)$$

where $UpSample$ is the up-sampling operation. Skip connections will be denoted as in Equation 10,

$$S_1 = Concat(U_1, F_1) \quad (10)$$

This step combines features from the up-sampled output and the corresponding feature map from the encoding stage. The decoder convolution will be denoted as in Equation 11,

$$F_2 = Conv(S_1, W_2) + b_2 \quad (11)$$

The output convolution is shown in Equation 12,

$$Y = Conv(F_1, W_{out}) + b_{out} \quad (12)$$

Table 2 Layers of the Unet model with ASPP module

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, 256, 256, 3)]	0
conv2d_1 (Conv2D)	(None, 256, 256, 64)	1792

conv2d_2 (Conv2D)	(None, 256, 256, 64)	36928
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 64)	0
conv2d_3 (Conv2D)	(None, 128, 128, 128)	73856
conv2d_4 (Conv2D)	(None, 128, 128, 128)	147584
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 128)	0
conv2d_5 (Conv2D)	**(None, 64, 64, 256)**	**295168**
conv2d_6 (Conv2D)	**(None, 64, 64, 256)**	**590080**
conv2d_7 (Conv2D)	**(None, 64, 64, 256)**	**590080**
conv2d_transpose_1 (Conv2DTranpose)	(None, 128, 128, 128)	131200
concatenate_1 (Concatenate)	(None, 128, 128, 256)	0
conv2d_8 (Conv2D)	(None, 128, 128, 128)	295040
conv2d_9 (Conv2D)	(None, 128, 128, 128)	147584
conv2d_transpose_2 (Conv2DTraspose)	(None, 256, 256, 64)	32832
concatenate_2 (Concatenate)	(None, 256, 256, 128)	0
conv2d_10 (Conv2D)	(None, 256, 256, 64)	73792
conv2d_11 (Conv2D)	(None, 256, 256, 64)	36928
conv2d_12 (Conv2D)	(None, 256, 256, 10)	650
Total params: 2,453,514 Trainable params: 2,453,514 Non-trainable params: 0		

Skip connections, similar to the UNet architecture, are used to bridge the encoder and decoder blocks. These connections help in preserving spatial information and gradients during the upsampling process, aiding in accurate localization of objects and maintaining fine details. Table 2 gives the model summary of the above implemented model with various dilation rates such as 2,4 and 6.

4.d. Proposed Pre-Trained Models for Feature extraction

4.d.1 Proposed Customized Model

The convolution layer which extracts features by input image while preserving the relationship between image pixels. In this layer edge detection, blur, sharpen filters can be applicable to the input image. At each area on the input, matrix multiplication is performed by consolidating the outcome. The resultant feature map is explained as:

$$P_a^r = \frac{P_a^{r-1} - Q_a^r}{T_a^r} + 1 \quad (13)$$

$$P_b^r = \frac{P_b^{r-1} - Q_b^r}{T_b^r} + 1 \quad (14)$$

Equation 13 and Equation 14 is the last layer's output feature map's width and height, where (P_a, P_b) is the dimension and height of the output feature map of the final layer and (Q_a, Q_b) is the kernel size, (T_a, T_b) that characterizes the total pixels kept away by the kernel in the r index and both the horizontal and vertical axes demonstrates the layer i.e., r = 1. Convolution operation is characterized as:

$$A_1 (c , d) = (I * J)(c , d) \quad (15)$$

In Equation 15, A1 (*c, d*) is a coplanar output feature map acquired by convolving the two-dimensional kernel *R* of size (*Qa, Qb*) and input feature map *I*. The sign * is utilized to address the convolution between *I* and *J*. The convolution operation is indicated as,

$$A_1(c , d) = \sum_{p=\frac{-L_x}{2}}^{P=\frac{+L_x}{2}} \cdot \sum_{q=\frac{-L_y}{2}}^{q=\frac{+L_y}{2}} \cdot I (c - p , d - q) J (p , q) \quad (16)$$

In Equation 16, A1 (c, d) is a final convolution operation of feature map where there is a convolution between the I and J. When the image is huge, pooling layers are frequently used to reduce the number of parameters. To minimise the image's dimensions, it either downsamples or upsamples. There are three different types of pooling: maximum, average, and sum. Max Pooling is mainly used to select the largest element from a feature map. Table 3 shows the layers created in customized model. Average pooling and dropout are used at appropriate places to reduce the number of parameters.

Table 3 Layers of the customized model

Layer (type)	**Output Shape**	**Param #**
conv2d_10 (Conv2D)	(None, 222, 222, 32)	896
average_pooling2d_2	Average (None, 111, 111, 32)	0
batch_normalization_5	Batch (None, 111, 111, 32)	128
conv2d_11 (Conv2D)	(None, 109, 109, 64)	18496
conv2d_12 (Conv2D)	(None, 107, 107, 64)	36928

average_pooling2d_3	(Average (None, 53, 53, 64)	0
batch_normalization_6	(Batch (None, 53, 53, 64)	256
conv2d_13 (Conv2D)	(None, 51, 51, 128)	73856
conv2d_14 (Conv2D)	(None, 49, 49, 128)	147584
conv2d_15 (Conv2D)	(None, 47, 47, 128)	147584
max_pooling2d_2	(MaxPooling2 (None, 24, 24, 128)	0
batch_normalization_7	(Batch (None, 24, 24, 128)	512
conv2d_16 (Conv2D)	(None, 22, 22, 256)	295168
conv2d_17 (Conv2D)	(None, 20, 20, 256)	590080
conv2d_18 (Conv2D)	(None, 18, 18, 256)	590080
conv2d_19 (Conv2D)	(None, 16, 16, 256)	590080
max_pooling2d_3	(MaxPooling2 (None, 8, 8, 256)	0
batch_normalization_8	(Batch (None, 8, 8, 256)	1024
flatten_4 (Flatten)	(None, 16384)	0
dense_15 (Dense)	(None, 64)	1048640
batch_normalization_9	(Batch (None, 64)	256
dropout_13 (Dropout)	(None, 64)	0
dense_16 (Dense)	(None, 10)	650
Total params: 3,542,218 **Trainable params: 3,541,130** **Non-trainable params:1,088**		

Pooling layers are commonly utilized to minimize the number of parameters when the image is large. It either uses upsampling or downsampling to reduce dimensionalities of the image. Then batch normalization [38] is used to make the data flow between intermediate layers of the neural network by making use of a higher learning rate.

4.d.2. Proposed Hybrid Model:

The customized architecture is fused with the AlexNet architecture as well as with the ResNet architecture and EfficientNet B7. In the working model of the hybrid model, after being resized fed into UNet for segmentation and the segmented images are passed through the concatenated CNN model. Here the output layer of the first model is given as input to the second model thus concatenating two models to form hybrid architecture. The model hybridization is taken into account in the view of increasing the accuracy. After passing through the hybridized model, the images are classified.

4.e Hyper parameter tuning

Hyper parameter tuning is utilized for expanding the productivity of the model. It does its little job to work on the accuracy of a model. The modification process holds significant importance, as certain alterations impact factors such as training computation time, convergence speed, and the utilization of processing units. Iterative hyperparameter tuning [45] was consistently performed to enhance the precision of the proposed model and improve overall accuracy. Table 4 represents the parameters for the proposed models. For the experiment, the training is done for each CNN architecture along with the hyper parameter tuning. In time of training, the tuning of the parameters is done for the reducing the dimensionality of features, regularization for improvement capacities on the result layer respectively. During the network training, batch size of 128 for the training data and batch size of 5 for the validation data, number of epochs being 50 with a 0.001 of learning rates is employed.

Table 4 Parameters of the Proposed Models

Parameter	Value
Batch size (Training data)	128
Batch size (Validation data)	5
Epoch	50
Optimizer	ADAM
Learning rate	0.001

4.f Weight-based Pruning

Weight-based pruning reduces the number of parameters in a neural network by removing small weights that have minimal impact on the model's output. This simplifies the model, reducing memory usage and computation time. The aim is to prune weights in a way that maintains the model's performance, with any performance loss being recoverable through fine-tuning [46][47].

Step 1: Train the Customized Model: Begin by training the model on the plant disease dataset. Use the customized CNN with UNet and ASPP integrated with the segmentation algorithms to achieve high accuracy.

Step 2: Compute the Magnitude of Weights: After training, compute the absolute magnitude of each weight in the network. For every weight W_{ij} compute its magnitude as: $|W_{ij}|$. Weights with smaller magnitudes contribute less to the model's output, and hence, are candidates for pruning.

Step 3: Define Pruning Threshold: A pruning threshold δ is defined based on the desired sparsity level of the model. In this experiment it is started with 0.05 and increased upto 0.20 without affecting the performance. Weights with magnitudes smaller than δ are considered insignificant and are pruned (set to zero).

Step 4: Prune Weights: Prune the weights that meet the condition This effectively removes the less significant weights from the network, making the model sparser.

Prune if: $|W_{ij}| < \delta$

Step 5: Fine-Tune the Pruned Model: After pruning, the model needs to be fine-tuned to recover any potential performance loss due to the removal of weights. The fine-tuning process involves re-training the pruned model for a few more epochs on the same dataset.

5. EXPERIMENTATION, RESULTS AND DISCUSSION

5.1 Experimental Setup

In order to evaluate the efficacy of the proposed classification and segmentation technique, six frequently used assessment metrics—namely accuracy, sensitivity, specificity, precision, and recall—are employed. The terms True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are designated accordingly. Accuracy is defined as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{17}$$

The proportion of true positive result to real positive is called the sensitivity in detecting disease and it is calculated as follows

$$Sensitivity = \frac{TP}{TP + FN} \tag{18}$$

The proportion of negative results to all real negatives is called Specificity or True Negative Rate (TNR). If the specificity is high, it indicates that the model is good in identifying the healthy leaves.

$$Specificity = \frac{TN}{TN + FP} \tag{19}$$

The ratio of positive inspections that are anticipated accurately to the total positively anticipated inspections is called Precision.

$$Precision\ factor = \frac{TP}{TP + FP} \tag{20}$$

The ratio of positive inspections that are anticipated accurately to all the inspections in actual class is called Recall.

$$Recall = \frac{TP}{TP + FN} \tag{21}$$

The Intersection over Union (IoU) or Jaccard Index is a metric used to evaluate the accuracy of segmentation. In this context, TP (True Positive) denotes the count of pixels accurately classified as foreground (correctly segmented), FP (False Positive) indicates the count of pixels erroneously classified as foreground (over-segmented), and FN (False Negative) denotes the count of pixels inaccurately classified as background (under-segmented).

$$IoU = \frac{TP}{TP + FP + FN} \tag{22}$$

5.2 Experimental Parameters

The first experiment is to compare the results of the non-segmented images with the ResNet50, AlexNet, EfficientNet B0, EfficientNet B7, customized and Hybrid models.

Table 5 Classification Model performances with Non-Segmented Images

Deep Learning Architectures	**Parameters (in Millions)**	**Training Time (in hours)**	**Training Accuracy**	**Validation Accuracy**
ResNet - 50	*23.5 M*	*15.2*	*97.0*	*93.5*
AlexNet	*58.3 M*	*25.3*	*92.9*	*91.2*
EfficientNet B0	*5.3M*	*7.5*	*93.9*	*92.9*
EfficientNet B7	*66.2M*	*28.6*	*98.0*	*95.8*
Customised Model	***3.54 M***	***3.3***	***98.2***	***96.9***
HybridResNet-50 with Customised Model	*85.4 M*	*35.5*	*94.1*	*91.6*

Hybrid AlexNet with Customised Model	*100.7 M*	*10.5*	*93.5*	*93.3*
Hybrid EfficientNet B7 with Customised Model	***72.3 M***	***30.3***	***98.9***	***98.5***

These architectures are trained with plant village dataset and tested by test data images of the same. It is also compared by changing weights of these models. The results of the experiments are presented in Table 5, revealing that the tailored model incorporating EfficientNet achieves superior accuracy and lower loss. Notably, the customized model not only demonstrates enhanced accuracy but also exhibits a notably reduced time requirement for each step.

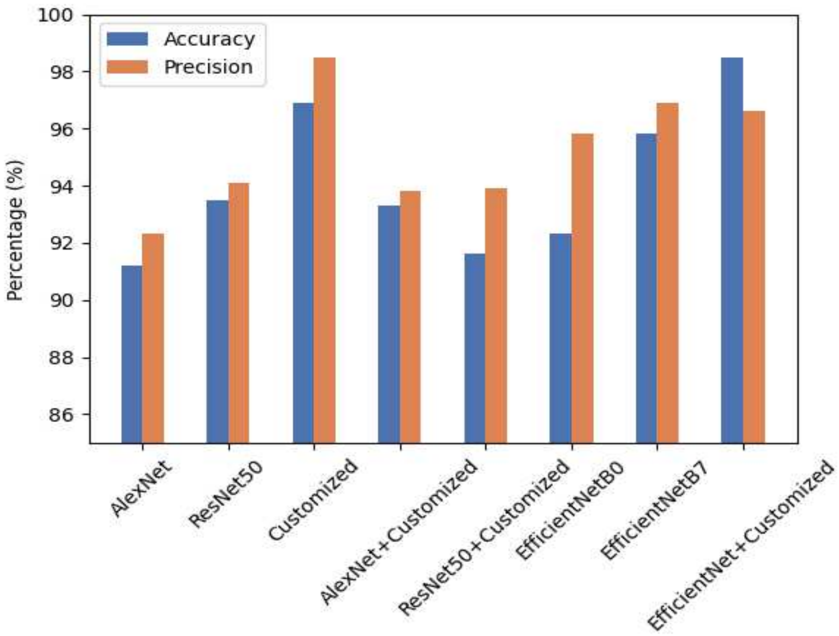


Fig.4a Accuracy and Precision of the Models with Non-Segmented Images

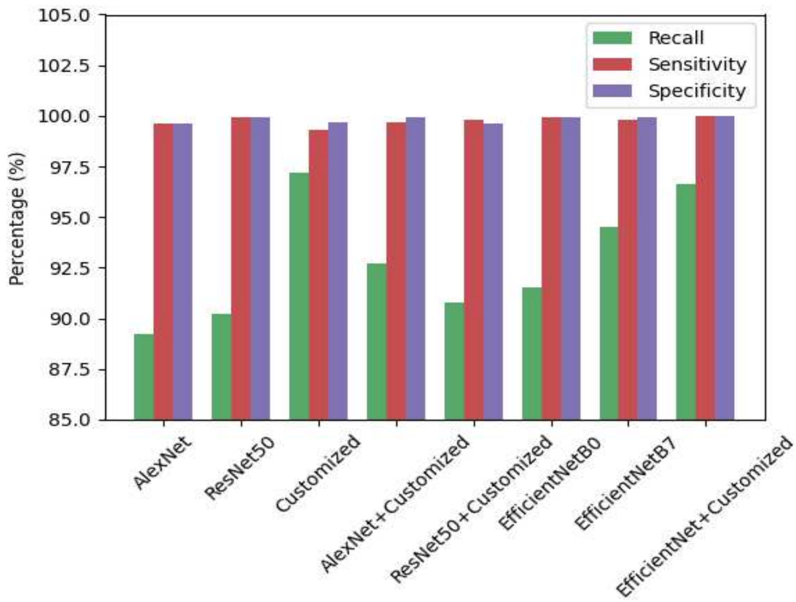


Fig.4b Recall, Sensitivity and Specificity of the Models with Non-Segmented Images

Figure 4a and 4b illustrates a performance metrics comparison, encompassing accuracy, precision, recall, sensitivity, and specificity, between the predefined and customized architectures. The next experiment is to segment the images using segmentation architectures like UNet, DeeplabV3+, iterative UNet and UNet with ASPP module.

The following table 6 shows the segmentation performance of each model for the plant village dataset.

Table 6 Model performances of Segmentation Architectures

Metrics	**DeepLabV3+**	**UNet**	**Iterative UNet**	**UNet with ASPP**
Accuracy	0.70	0.74	0.90	**0.86**
Sensitivity	0.83	0.82	0.86	0.88
Jaccard(IOU)	**0.89**	0.72	0.87	**0.92**
Precision	0.74	0.68	0.80	0.89

From the table 6 and figure 5 it is inferred that, U-Net tends to perform well in terms of accuracy because of its encoder-decoder structure with skip connections. The skip connections allow for the fusion of both low-level and high-level features, enabling the model to capture detailed spatial information during segmentation. As a result, U-Net can effectively delineate object boundaries and generate accurate pixel-level predictions.

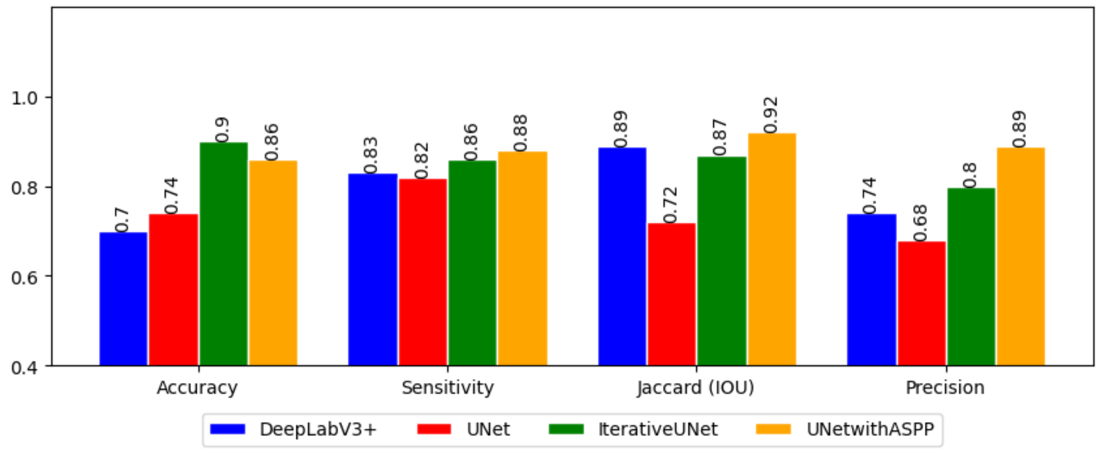


Fig.5 Performance of the Segmentation Algorithms

Whereas DeepLabv3, on the other hand, focuses on capturing multi-scale contextual information through the use of dilated convolutions and ASPP. By incorporating dilated convolutions, DeepLabv3 can aggregate information from a wider receptive field, which helps in understanding global context and handling objects of varying scales. ASPP further enhances this capability by performing convolutions at multiple dilation rates. The Jaccard IOU metric, which measures the overlap between predicted and ground truth segmentation masks, evaluates the model's ability to capture object shapes and contours accurately.

Thus, the proposed segmentation models Iterative UNet and Unet with ASPP outperform the above traditional models in the following ways. Iterative U-Net aims to enhance accuracy, Jaccard IOU, precision, and sensitivity through an iterative feedback loop that refines the segmentation predictions. By iteratively updating the predictions based on the input image and previous iterations, iterative U-Net aims to improve segmentation performance in terms of these evaluation metrics. Whereas in the other model UNet with ASPP module, ASPP is incorporated to UNet architecture similar to DeepLabV3+. In this way, it helps to capture contextual information at multiple levels. By considering different dilation rates in the ASPP module, the model can effectively analyze the image at multiple scales and gather contextual cues. This enables the proposed model to have a more comprehensive understanding of the scene, distinguishing true object boundaries from background clutter and reducing the likelihood of false positive predictions.

The third experiment is to compare the results of the segmented images using UNet with ASPP with the classification models discussed. The inference obtained from Table 7 is that Hybrid EfficientNet B7 with customized Model gave a better accuracy when comparing to other models.

But the tradeoff between time and accuracy shows that customized model with pruning is producing a nearby accuracy as like hybrid models as well as with reduced time complexity.

Table 7 Model performances with Segmented Images using UNet with ASPP

Deep Learning Architectures	**Parameters (in Millions)**	**Training Time (in hours)**	**Training Accuracy**	**Validation Accuracy**
ResNet - 50	*23.5 M*	*11.2*	94.79	*95.5*
AlexNet	*58.3 M*	*15.3*	93.31	*92.8*
EfficientNet B0	*5.3M*	*4.5*	*94.1*	*93.1*
EfficientNet B7	*66.2M*	*16.3*	*98.7*	*96.8*
Customised Model	***3.54 M***	***2.6***	***98.9***	***97.5***
ResNet-50 with Customised Model	*85.4 M*	*26.4*	*97.8*	*97.4*
Hybrid AlexNet with Customised Model	*100.7 M*	*8.6*	*94.64*	*94.6*
Hybrid EfficientNet B7 with Customised Model	***72.3 M***	***18.3***	***99.6***	***99.7***
Customised Model with pruning	***2.31 M***	***1.2***	***98.9***	***97.5***

Figure 6 is a bar graph representation that shows the comparison of performance of the pre-trained models for the segmented images based on performance metrics like accuracy, precision, recall etc. Comparing with Figure 6, it is evident that the classification accuracy shows a consistent increase while the time complexity decreases when performing classification after image segmentation.

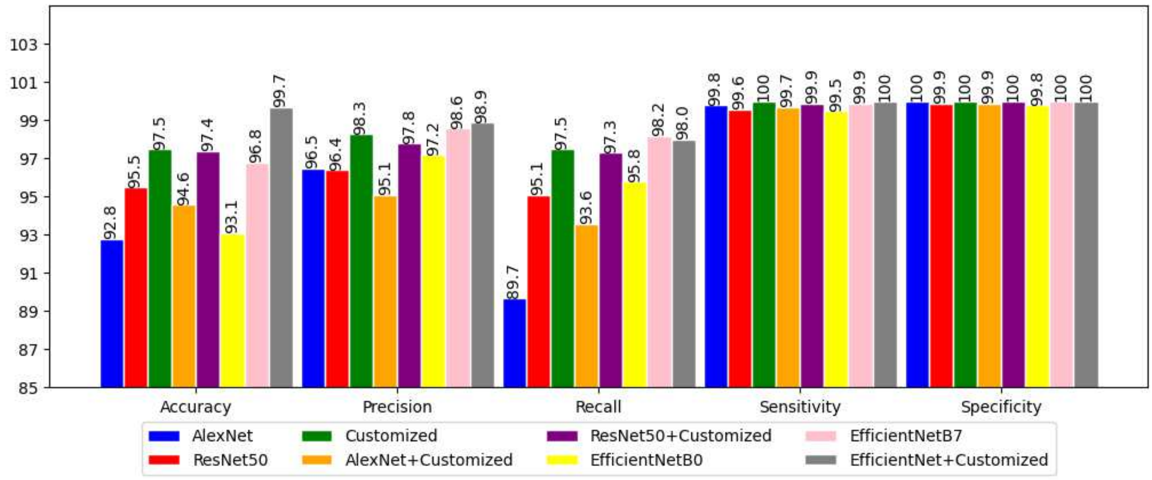


Fig.6 Performance of the Models with segmented Images

In the fourth experiment, the application of different optimizers is explored on the customized model. The result shown in Table 8 and Table 9 indicates that a pruned customized model with Adam optimizer has higher accuracy as well as lower loss.

Table 8 Pruned Customized Model performance with different optimizers without segmentation

Optimisers	**Training Accuracy**	**Validation Accuracy**	**Precision**	**Recall**	**Specificity**	**Sensitivity**
Adam	*99*	*98.8*	*97.9*	*97.6*	*100*	*99.8*
Adamax	*98*	*96.4*	*97.5*	*97*	*100*	*100*
Adagrad	*96.96*	*96.07*	*97.76*	*96.74*	*100*	*100*
RMSprop	*96.85*	*91.36*	*92.97*	*90.56*	*99.66*	*99.66*

In the final testing phase, 20 images were utilized to represent 10 distinct classes. Throughout this testing process, the models put forth exhibited an impressive testing accuracy of 97%. It is noteworthy that the ResNet50 model emerged with the highest success rate, accompanied by lower losses and convergence times in comparison to models employing alternative pre-trained networks.

The results are obtained for the crop leaf diseases of Corn, Tomato and Potato from the PlantVillage dataset. The first experiment is designed to analyze the outcomes of the non-segmented images with the CNN models and Hybrid models. The second experiment is to compare the results of segmented images with the CNN models and the hybrid model.

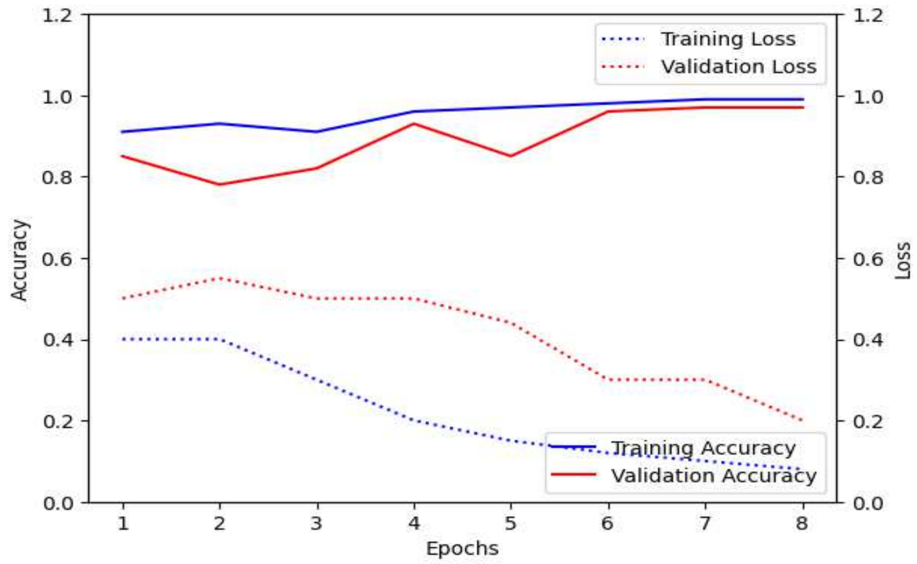
Table 9 Model performance for the Pruned Customized Model with respect to optimizers with segmentation

Optimisers	Training Accuracy	Validation Accuracy	Precision	Recall	Specificity	Sensitivity
Adam	*99.9*	***99.8***	*98.3*	*98.1*	*100*	*100*
Adamax	*98.04*	*98.04*	*98.25*	*97.81*	*100*	*99.97*
Adagrad	*95.94*	*96.74*	*97.41*	*96.01*	*100*	*100*
RMSprop	*95.22*	*91.56*	*93.16*	*89.63*	*99.73*	*99.9*

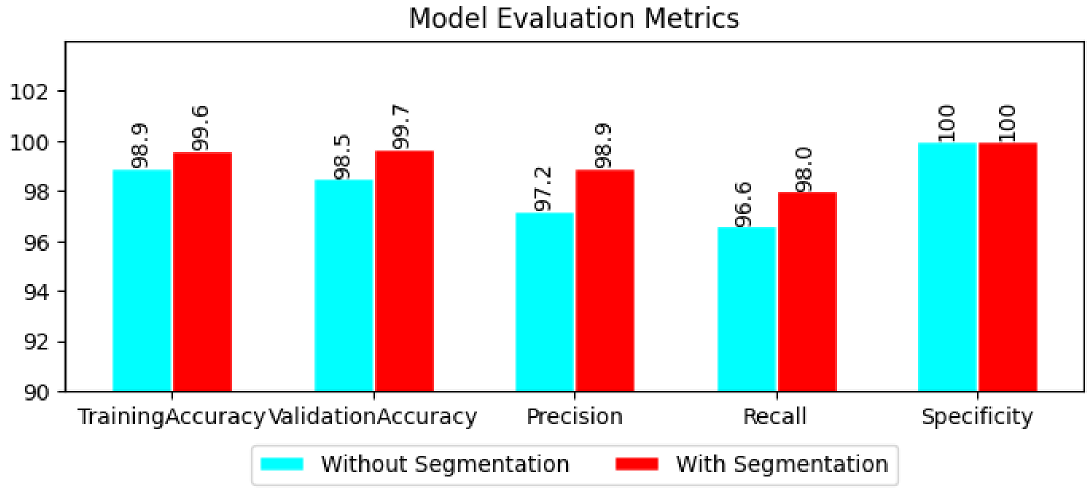
4.3 Graphical representation of performance of architectures

4.3.1 Graphical representation of Customized model+EfficientNet B7

Figure 7 (a) shows the training and validation, accuracy and loss curves of customized model when concatenated with EfficientNet B7. Figure 7 (b) shows the performance improvement of the concatenated customized model with segmentation using UNet with ASPP architecture.



(a)

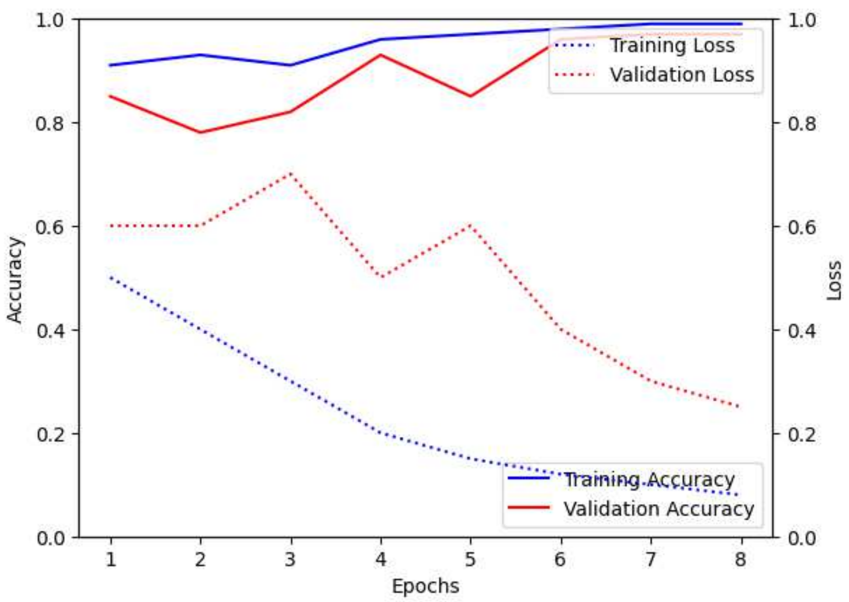


(b)

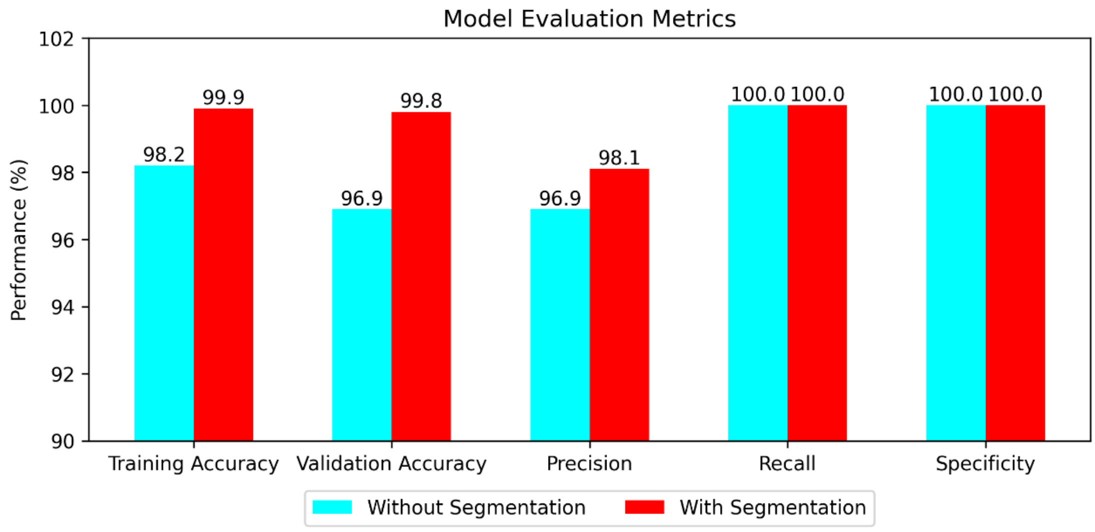
Fig.7 Performance plots of Customized Model with EfficientNet B7 (a) accuracy and loss with segmentation (b) comparison with and without segmentation

4.3.1 Graphical representation of Pruned Customized model

Figure 8 (a) shows the training and validation, accuracy and loss curves of customized model. Figure 8 (b) shows the performance improvement of the customized model with segmentation using UNet with ASPP architecture.



(a)



(b)

Fig.8 Performance plots of Pruned Customized Model (a) accuracy and loss with segmentation (b) comparison with and without segmentation

4.4. Comparison with State of the Art Methods

Table 10 shows the validation accuracy attained by pre-trained, customized and hybrid models by using segmented and non-segmented images.

Table 10 Comparison of the proposed work

Methods	**Dataset**	**Accuracy**		
		Model name	**With segmentation**	**Without segmentation**
Segmentation, Feature Extraction	Plant Village (Corn, potato and tomato leaves)	ResNet - 50	*95.5*	*93.5*
		AlexNet	*92.8*	*91.2*
		EfficientNet B0	*93.1*	*92.9*
		EfficientNet B7	*96.8*	*95.8*
		ResNet-50 with Customised Model	*97.4*	*91.6*
		Hybrid AlexNet with Customised Model	*94.6*	*93.3*
		Hybrid EfficientNet B7 + Customised Model	***99.7***	***98.5***
		Pruned Customised Model	***99.8***	***96.9***

Table 11 plays a pivotal role in encapsulating the essential references for the proposed work, meticulously outlining the various models, methodologies, and datasets that were leveraged in the referenced studies. This comprehensive compilation serves as a rich repository of insights, offering a nuanced perspective on the underpinnings of the research.

Upon a closer examination of the comparative results delineated in Table 11, a discernible trend emerges. It becomes evident that the amalgamation of a customized architecture with the incorporation of pretrained models, such as the notable EfficientNet B7, alongside an adept segmentation process, yielded superior accuracy outcomes when contrasted with the standalone pretrained models. This nuanced observation underscores the efficacy of the proposed approach, highlighting the synergistic impact of a tailored architecture in conjunction with state-of-the-art pretrained models. The meticulous fusion of these components not only enhances accuracy but also signifies a noteworthy stride towards optimizing model performance, thus validating the strategic choices made in the course of this research endeavour.

Table 11 State of the art comparison

S.No	Author	Reference No	Methods	Models and Algorithm	Dataset	Accuracy
1	KhaoulaBelhajSoulami	[21]	Segmentation	UNet	DDSM, Inbreast	90.5%
2	YonghuaXiong	[22]	Segmentation	AISA based on GrabcutAlgorithm, MobileNet	PlantVillage + expanded dataset	80%
3	Muhammad Attique Khan	[23]	Coefficient based segmentation, Feature Fusion	VGG16, caffeAlexnet, Multi class SVM	Plant Village, CASC-IFW	98.60%.
4	T Daniya	[24]	Segmentation	SegNet, Deep RNN, RSW algorithm	Rice Detection Dataset	90.5%
5	Shuo Chen	[25]	Segmentation	BLS model based on UNet	Own Dataset	98.2%
6	Md. Arifur Rahman	[26]	Segmentation	RGB threshold	Plant Village	99.25%

7	YahyaAltunta	[27]	Feature Extraction, Concatenation	AlexNet, GoogLeNet and ResNet-50	PlantVillage	96.99%.
8	Faye Mohameth	[28]	Feature Extraction	ResNet 50, Google Net, VGG-16	PlantVillage	99.53%
9	Vinod Kumar	[29]	Feature Extraction	ResNet34	open image dataset	99.40%
10	Malliga Subramanian	[30]	Transfer learning and hyper- parameters optimization	VGG16, ResNet50, InceptionV3, and Xception	PlantVillage	93%
11	Mobeen-ur-Rehman	[31]	Feature Extraction	Customised CNN model	MESSIDOR	98.94%
12	ShradhaVerma	[32]	Transfer learning and Feature extraction	AlexNet, SqueezeNet and Inception V3	PlantVillage	93.4%
13	Chunshan Wang	[33]	Segmentation,	DeepLabV3 ,UNet	Self Collected Dataset	92.85%
14	Bandar Alotaibi	[34]	Hybrid Approach	hybrid deep ResNet-Inception	Pavia University, Pavia Centre scenes , Salinast andIndian Pines datasets.	99.02%
15	Yangyang Li	[35]	Deep Feature Fusion	Deep ResNet50 and VGG16	UCM, NWPURESISC45	94.03%
16	Mohan M	[44]	Hybrid Approach	Unet+k-means and kookaburra optimization algorithm	plant village +plant leaves for image classification + new plant diseases dataset	98%
17	Mohana Saranya	**Propose d Work**	Hybrid Approach, Segmentation, Classification	**Pruned Customised Model**	Plant Village (Corn, potato and tomato leaves)	***99.8***
				Hybrid EfficientNet B7 with Customised Model		***99.7***

6. CONCLUSION AND FUTURE SCOPE

In this study, both the proposed pruned customized model and its combination with the pretrained EfficientNet B7 architecture have yielded highly promising results in the realm of classification. Notably, the application of segmentation to the input dataset, utilizing the UNet with ASPP approach before feeding it into the classification model, further elevated accuracy levels. The proposed methodology has demonstrated superior performance compared to existing approaches, exhibiting higher accuracy and reduced number of parameters. The incorporation of segmentation techniques played a pivotal role in boosting accuracy when compared to non-segmented approaches. Additionally, the concatenated model, which fuses the strengths of individual models, showcased superior performance, emphasizing the synergistic benefits of combining segmentation and classification processes. Thus, the proposed approach emerges as a robust and effective solution, excelling in both performance and accuracy metrics.

The utilization of the plant village dataset for model training and evaluation underscores the real-world applicability of the approach in the context of diverse plant diseases across multiple species. Looking ahead, future research endeavors could explore extending this approach to datasets featuring an increased number of classes and diverse compositions. Additionally, investigating methods to optimize hyperparameters could further enhance the practicality and scalability of the proposed methodology. In summary, the study not only establishes the effectiveness of the proposed pruned customized models and segmentation techniques but also positions them as advanced solutions that outperform existing approaches in the critical task of plant disease classification.

Abbreviations

DL- Deep Learning

CNN - Convolutional Neural Network

RNN - Recurrent Neural Networks

ASPP - Atrous Spatial Pyramid Pooling

IOU- Intersection over Union

DDSM - Digital Database for Screening Mammography

Availability of data and material

The manuscript data set can be freely downloaded from https://www.tensorflow.org/datasets/catalog/plant_village

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We extend our appreciation to Kongu Engineering College for facilitating access to state-of-the-art GPU facilities, enhancing the computational capabilities crucial for data analysis and simulations.

REFERENCES

[1] S. Vetal and K. R.S., "Tomato Plant Disease Detection using Image Processing," *Ijarcce*, vol. 6, no. 6, pp. 293–297, 2017, doi: 10.17148/ijarcce.2017.6651.

[2] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, no. September 2017, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[3] J. Wang and X. Liu, "Medical image recognition and segmentation of pathological slices of gastric cancer based on Deeplab v3+ neural network," *Comput. Methods Programs Biomed.*, vol. 207, p. 106210, 2021, doi: 10.1016/j.cmpb.2021.106210.

[4] L. Deng, S. Liu, X. Wang, G. Zhao, and J. Xu, "Particle Swarm Optimization and Salp Swarm Algorithm for the Segmentation of Diabetic Retinal Blood Vessel Images," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–14, 2022, doi: 10.1155/2022/1936482.

[5] J. Czajkowska, P. Badura, S. Korzekwa, and A. Płatkowska-Szczerek, "Automated segmentation of epidermis in high-frequency ultrasound of pathological skin using a cascade of DeepLab v3+ networks and fuzzy connectedness," *Comput. Med. Imaging Graph.*, vol. 95, no. December 2021, 2022, doi: 10.1016/j.compmedimag.2021.102023.

[6] M. N. Ab Wahab, A. Nazir, A. T. Z. Ren, M. H. M. Noor, M. F. Akbar, and A. S. A. Mohamed, "Efficientnet-Lite and Hybrid CNN-KNN Implementation for Facial Expression Recognition on Raspberry Pi," *IEEE Access*, vol. 9, pp. 134065–134080, 2021, doi: 10.1109/ACCESS.2021.3113337.

[7] X. Liang *et al.*, "Patch attention layer of embedding handcrafted features in CNN for facial expression recognition," *Sensors (Switzerland)*, vol. 21, no. 3, pp. 1–15, 2021, doi: 10.3390/s21030833.

[8] J. Cao, J. Zhang, and X. Jin, "A Traffic-Sign Detection Algorithm Based on Improved Sparse R-cnn," *IEEE Access*, vol. 9, pp. 122774–122788, 2021, doi: 10.1109/ACCESS.2021.3109606.

[9] C. jin Li, Z. Qu, S. ye Wang, and L. Liu, "A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment," *Pattern Recognit. Lett.*, vol. 145, pp. 127–134, 2021, doi: 10.1016/j.patrec.2021.02.003.

[10] T. Liang, H. Bao, W. Pan, and F. Pan, “Traffic Sign Detection via Improved Sparse R-CNN for Autonomous Vehicles,” *J. Adv. Transp.*, vol. 2022, 2022, doi: 10.1155/2022/3825532.

[11] T. Khan, R. Sarkar, and A. F. Mollah, *Deep learning approaches to scene text detection: a comprehensive review*, vol. 54, no. 5. Springer Netherlands, 2021.

[12] K. Boukthir, A. M. Qahtani, O. Almutiry, H. Dhahri, and A. M. Alimi, “Reduced annotation based on deep active learning for arabic text detection in natural scene images,” *Pattern Recognit. Lett.*, vol. 157, pp. 42–48, 2022, doi: 10.1016/j.patrec.2022.03.016.

[13] J. Liu and X. Wang, “Tomato Diseases and Pests Detection Based on Improved Yolo V3 Convolutional Neural Network,” *Front. Plant Sci.*, vol. 11, no. June, pp. 1–12, 2020, doi: 10.3389/fpls.2020.00898.

[14] A. Gutierrez, A. Ansuategi, L. Susperregi, C. Tubío, I. Rankić, and L. Lenža, “A Benchmarking of Learning Strategies for Pest Detection and Identification on Tomato Plants for Autonomous Scouting Robots Using Internal Databases,” *J. Sensors*, vol. 2019, 2019, doi: 10.1155/2019/5219471.

[15] J. P. Gonçalves, F. A. C. Pinto, D. M. Queiroz, F. M. M. Villar, J. G. A. Barbedo, and E. M. Del Ponte, “Deep learning architectures for semantic segmentation and automatic estimation of severity of foliar symptoms caused by diseases or pests,” *Biosyst. Eng.*, vol. 210, pp. 129–142, 2021, doi: 10.1016/j.biosystemseng.2021.08.011.

[16] L. G. Divyanth, A. Ahmad, and D. Saraswat, “A two-stage deep-learning based segmentation model for crop disease quantification based on corn field imagery,” *Smart Agric. Technol.*, vol. 3, no. January 2022, p. 100108, 2023, doi: 10.1016/j.atech.2022.100108.

[17] P. Bedi and P. Gole, “Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network,” *Artif. Intell. Agric.*, vol. 5, pp. 90–101, 2021, doi: 10.1016/j.aiia.2021.05.002.

[18] V. Sravan, K. Swaraj, K. Meenakshi, and P. Kora, “A deep learning based crop disease classification using transfer learning,” *Mater. Today Proc.*, no. xxxx, 2021, doi: 10.1016/j.matpr.2020.10.846.

[19] B. Li and D. Lima, “Facial expression recognition via ResNet-50,” *Int. J. Cogn. Comput. Eng.*, vol. 2, no. January, pp. 57–64, 2021, doi: 10.1016/j.ijcce.2021.02.002.

[20] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, “Deep learning in multi-object detection and tracking: state of the art,” *Appl. Intell.*, vol. 51, no. 9, pp. 6400–6429, 2021, doi: 10.1007/s10489-021-02293-7.

[21] K. B. Soulami, N. Kaabouch, M. N. Saidi, and A. Tamtaoui, “Breast cancer: One-stage automated detection, segmentation, and classification of digital mammograms using UNet model based-semantic segmentation,” *Biomed. Signal Process. Control*, vol. 66, no. December 2020, p. 102481, 2021, doi: 10.1016/j.bspc.2021.102481.

[22] Y. Xiong, L. Liang, L. Wang, J. She, and M. Wu, “Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset,” *Comput. Electron. Agric.*, vol. 177, no. July, p. 105712, 2020, doi: 10.1016/j.compag.2020.105712.

[23] M. A. Khan *et al.*, “CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features,” *Comput. Electron. Agric.*, vol.

155, no. February, pp. 220–236, 2018, doi: 10.1016/j.compag.2018.10.013.

[24] T. Daniya and S. Vigneshwari, “Deep Neural Network for Disease Detection in Rice Plant Using the Texture and Deep Features,” *Comput. J.*, vol. 00, no. 00, 2021, doi: 10.1093/comjnl/bxab022.

[25] S. Chen *et al.*, “An approach for rice bacterial leaf streak disease segmentation and disease severity estimation,” *Agric.*, vol. 11, no. 5, 2021, doi: 10.3390/agriculture11050420.

[26] M. A. Rahman, M. M. Islam, G. M. S. Mahdee, and M. W. Ul Kabir, “Improved Segmentation Approach for Plant Disease Detection,” *1st Int. Conf. Adv. Sci. Eng. Robot. Technol. 2019, ICASERT 2019*, vol. 2019, no. Icasert, pp. 1–5, 2019, doi: 10.1109/ICASERT.2019.8934895.

[27] Y. ALTUNTAŞ and F. KOCAMAZ, “Deep Feature Extraction for Detection of Tomato Plant Diseases and Pests based on Leaf Images,” *Celal Bayar Üniversitesi Fen Bilim. Derg.*, vol. 17, no. 2, pp. 145–152, 2021, doi: 10.18466/cbayarfbe.812375.

[28] F. Mohameth, C. Bingcai, and K. A. Sada, “Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village,” *J. Comput. Commun.*, vol. 08, no. 06, pp. 10–22, 2020, doi: 10.4236/jcc.2020.86002.

[29] V. Kumar, H. Arora, Harsh, and J. Sisodia, “ResNet-based approach for Detection and Classification of Plant Leaf Diseases,” *Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020*, no. Icesc, pp. 495–502, 2020, doi: 10.1109/ICESC48915.2020.9155585.

[30] M. Subramanian, K. Shanmugavadivel, and P. S. Nandhini, “On fine-tuning deep learning models using transfer learning and hyper-parameters optimization for disease identification in maize leaves,” *Neural Comput. Appl.*, vol. 0123456789, 2022, doi: 10.1007/s00521-022-07246-w.

[31] Mobeen-Ur-Rehman, S. H. Khan, Z. Abbas, and S. M. Danish Rizvi, “Classification of Diabetic Retinopathy Images Based on Customised CNN Architecture,” *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, pp. 244–248, 2019, doi: 10.1109/AICAI.2019.8701231.

[32] S. Verma, A. Chug, and A. P. Singh, “Application of convolutional neural networks for evaluation of disease severity in tomato plant,” *J. Discret. Math. Sci. Cryptogr.*, vol. 23, no. 1, pp. 273–282, 2020, doi: 10.1080/09720529.2020.1721890.

[33] C. Wang, P. Du, H. Wu, J. Li, C. Zhao, and H. Zhu, “A cucumber leaf disease severity classification method based on the fusion of DeepLabV3+ and U-Net,” *Comput. Electron. Agric.*, vol. 189, no. August, p. 106373, 2021, doi: 10.1016/j.compag.2021.106373.

[34] B. Alotaibi and M. Alotaibi, “A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification,” *PFG - J. Photogramm. Remote Sens. Geoinf. Sci.*, vol. 88, no. 6, pp. 463–476, 2020, doi: 10.1007/s41064-020-00124-x.

[35] Y. Li, Q. Wang, X. Liang, and L. Jiao, “A Novel Deep Feature Fusion Network For Remote Sensing Scene Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education , International Research Center for Intelligent Perception and Computation , Joint International Researc,” *IGARSS 2019 - 2019 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 5484–5487, 2019.

[36] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017, doi: 10.1109/TPAMI.2016.2644615.

[37] S. Sivagami, P. Chitra, G. S. R. Kailash, and S. R. Muralidharan, “UNet Architecture Based Dental Panoramic Image Segmentation,” *2020 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2020*, pp. 187–191, 2020, doi: 10.1109/WiSPNET48689.2020.9198370.

[38] A. Khamparia, G. Saini, D. Gupta, A. Khanna, S. Tiwari, and V. H. C. de Albuquerque, “Seasonal Crops Disease Prediction and Classification Using Deep Convolutional Encoder Network,” *Circuits, Syst. Signal Process.*, vol. 39, no. 2, pp. 818–836, 2020, doi: 10.1007/s00034-019-01041-0.

[39] Y. Yang, H. Shan, and F. Qu, "Maize disease segmentation method based on improved image segmentation network model." *2020 Int. Conf. Artificial Intelligence and Computer Engineering.*, vol. 12610 pp. 481-485, doi:10.1117/12.2671267.

[40] S. Talasila, K. Rawal, and G. Sethi, "Black gram disease classification using a novel deep convolutional neural network," *Multimedia Tools and Applications,* pp. 1-25, 2023, doi: 10.1007/s11042-023-15220-4.

[41] H. Shu, J. Liu, Y. Hua, J. Chen, S. Zhang, M. Su, et al., "A grape disease identification and severity estimation system," *Multimedia Tools and Applications,* pp. 1-18, 2023, doi: 10.1007/s11042-023-14755-w.

[42] L. Li, B. Wang, Y. Li, and H. Yang, "Diagnosis and mobile application of apple leaf disease degree based on a small-sample dataset," *Plants,* vol. 12, p. 786, 2023, doi: 10.3390/plants12040786.

[43] S. Mandal, “Survey on Plant Disease Detection using Deep Learning based Frameworks,” *International Journal of Medical, Pharmacy and Drug Research,* vol. 7, no. 2, 2023.

[44] M. M. Mohan and S. Anandamurugan, "Improving deep learning-based plant disease classification with convolutional block attention module (CBAM)," *Cognitive Computation*, vol. 17, no. 1, pp. 58-70, 2025.

[45] S. S. R. Depa, S. K. S. R. Anjaneyulu, and R. R. S. R. Reddy, "Using transfer learning-based plant disease classification and detection with ensemble learning," *BMC Plant Biology*, vol. 24, no. 1, pp. 1-15, 2024.

[46] Zhang, X., Li, Z., & Li, M, "A novel and efficient model pruning method for deep convolutional neural networks by evaluating the direct and indirect effects of filters," *Neurocomputing, 569*, 127124, 2024.

[47] Yeom, S.K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.-R., & Samek, W, "Pruning by explaining: A novel criterion for deep neural network pruning," *Pattern Recognition, 115*, 107899, 2021.

[48] S. E. K. Ebid, S. El-Tantawy, D. Shawky, and H. L. Abdel-Malek, "Correlation-based pruning algorithm with weight compensation for feedforward neural networks," Neural Computing and Applications, vol. 37, pp. 6351–6367, 2025.

[49] P. K. Das, S. S. Rupa, S. Pumrin, U. C. Das, and M. K. Hossen, "Deep Learning for Plant Disease

Detection and Classification: A Systematic Analysis and Review," Current Applied Science and Technology, vol. 24, no. 4, 2024.

[50] D. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, 2015, https://doi.org/10.48550/arXiv.1511.08060